Application No.: 10/047,809
Amendment dated: June 8, 2005
Reply to Office Action dated: March 8, 2005

AMENDMENTS TO THE SPECIFICATION

Please amend the following paragraphs:

On page 6, the paragraph beginning on line 24:

For example, in the Intel processor family, the Pentium® processor is an in-order machine with approximately three million transistors. By comparison, the Pentium® Pro, which is an out of order machine, uses about six and a half million transistors, requiring much more space than the in-order Pentium®. Because of the additional transistors, the Pentium® Pro also requires more power and generates more heat. ~~The present invention takes advantage of the existing~~ The existing space conserving design of the in-order ~~machine, which~~ machine is used in Intel's Itanium® Processor Family (IPF) by enabling the in-order platform to support a multi-threading processor.

On page 7, the paragraph beginning on line 13:

In-order multi-threading processor 12 ~~must execute~~ executes instructions in the order the instructions were entered into memory module 14. Therefore, unlike an out of order processor, in-order multi-threading processor 12 is unable to create independent chains of execution necessary to extract instruction level parallelism (ILP) from a single thread. To determine the dependencies of each of the instructions from the multiple threads, multi-threading computer system 10 relies on a specialized multi-thread scheduler and a compiler to identify sets of independent instructions and logic to schedule the threads.

On page 8, the paragraph beginning on line 22:

In one embodiment of the present invention, the compiler explicitly describes blocks of independent operations to the in-order machine so that may be executed in parallel. In contrast, a compiler for earlier machines was not capable of describing independent instructions. Instead, hardware was required to determine independent instructions at run time. Therefore, in one embodiment of the present invention, the task of generating instruction level parallelism is accomplished statically at compile time rather than dynamically at run time.

On page 9, the paragraph beginning on line 1:

This thread dispersal of the compiler ~~of the present invention~~ for in-order machines thus motivates the development of wide in-order machines that can execute many instructions simultaneously. In addition, to efficiently utilize the capabilities of a wide in-order machine, the machine must also be able to schedule multiple threads when the compiler cannot find enough ILP in a single thread to fully occupy the machine as described above with regard to multi-thread scheduler 24.

On page 10, the paragraph beginning on line 11:

Therefore, adding additional threads ~~to the present invention~~ could eliminate the advantage of multi-threading. In fact, the per-thread cost is even larger for an in-order machine than for an out-of-order machine. With the current configuration of in-order

multi-threading processor 12, it is not an efficient use of processing power to execute more than two threads at the same time, particularly because the processor is not currently wide enough to support more than two threads in parallel.